SPLINE PATH BASED AUTONOMOUS INFRASTRUCTURE INSPECTION USING AN UNMANNED AERIAL VEHICLE

A Thesis

by

LEELA KRISHNA C. GOTTUMUKKALA

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Robin R. Murphy
Committee Members,	Dezhen Song
	Le Xie
Head of Department,	Dilma Da Silva

December 2018

Major Subject: Computer Science

ABSTRACT

This research demonstrates a novel idea to autonomously conduct infrastructure inspection using an Unmanned Aerial Vehicle (UAV) via simulation and a proof of concept. Currently, inspection is conducted either manually, which is time-consuming and exhausting, or by using the Orbit mode, which results in the UAV either missing to capture views partially/completely of an irregularly structured (e.g., concave hull) target or having improper visibility of the target in the frame. The presented approach executes a part of the infrastructure inspection autonomously to capture a video of the target on a spline path with the orientation of the UAV's camera perpendicular to the course over ground and also the nadir views of the target. This allows infrastructure inspectors, first responders, insurance authorities to perform normal day-to-day or post-disaster inspections effectively, i.e. easily and with better and more target coverage. The performance of the presented algorithm was evaluated against orbit mode in simulation. From the simulated experiments, it was observed that using the presented algorithm the UAV's camera covered 6.2% ($p < 0.0027$) more of the target's perimeter, with 10.27° ($p < 4.95\text{x}10^{-6}$) closer to perpendicular observation of the target and 18.49 m ($p < 2.8\text{x}10^{-7}$) closer to target than in orbit mode. UAV faces the target for about 10.9% ($p < 0.0042$) duration less using the presented algorithm compared to orbit mode, while traveling 38.79 m ($p < 0.0102$) less and 12.2 s ($p < 0.0828$) faster. It was evident that the presented algorithm was better in terms of perimeter coverage, average deviation in the angle of depression w.r.t target, and average observation distance, while Orbit was better in terms of visual stability and video smoothness. A proof of concept for the presented algorithm was implemented and tested in real-world experimental test-cases using a 3DR-Solo. The implementation can be further expanded with modifications as future work, and is expected to outperform orbit mode for autonomous infrastructure inspection in all terms, especially in the case of irregularly structured targets, making the inspection process more effective, i.e. easy and with better and more target coverage.

DEDICATION

To Monkey D. Luffy, Gon Freecss, The Doctor and Tony Stark (the Stark that matters now).

ACKNOWLEDGMENTS

I wish to express my sincere appreciation to my family for allowing me to continue my education and for their support while working on this project. The success and outcome of this project required a lot of guidance and assistance, and I am extremely privileged to have got this all along for the completion of my project. All that I have done is only due to such supervision and assistance, and I would not forget to thank them.

I am grateful to Dr. Robin R. Murphy, my committee chairman, for her instructions, guidance and patience throughout this project. I would like to thank my fellow labmates: Jan Dufek, Grant Wilde, and Xuesu Xiao for their suggestions and opinions in helping me complete my class work. I would also like to thank Justin Adams, Jan Dufek, Grant Wilde, and Jesse Rosart-Brodnitz for their help in conducting experiments. Also, I would like to extend my sincere esteem to all staff in the department for their timely support.

Last but not least I am thankful to all of my friends who directly or indirectly helped me to complete my project.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a thesis (or) dissertation committee consisting of Professor Robin Murphy and Professor Dezhen Song of the Department of Computer Science and Associate Professor Le Xie of the Department of Electrical & Computer Engineering.

The data utilized for conducting simulation experiments were provided by CRASAR.

All other work conducted for the thesis (or) dissertation was completed by the student independently.

Funding Sources

Part of the Graduate study was supported by a Teaching Assistantship from Computer Science, Texas A&M University and a dissertation research fellowship from NSF.

NOMENCLATURE

CRASAR	Center for Robot-Assisted Search and Rescue
UAV	Unmanned Aerial Vehicle
GCS	Ground Control Station
ROI	Region Of Interest
RTH	Return To Home
FOV	Field Of Vision

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

During the aftermath of hurricane Irma, CRASAR Unmanned Aerial System (UAS) Team was requested to fly missions in support of post-disaster critical infrastructure assessments. The UAS Team flew 139 missions spread over five days, inspected around 100 targets and collected damage related images and video. Each inspection task consists of three phases - Image Capture phase, Video Capture phase, and Nadir Image Capture phase. In the Image Capture phase images of the target are captured using a UAV. In the Video Capture phase a continuous video of the target is captured by piloting the UAV around the target with its camera facing the target. In Nadir Image Capture phase images of the target are captured from above the target.

Currently, this task is executed by a combination of manual control, orbit mode, and auto flight mode. Orbit mode is sometimes used for the capturing video, using autonomous software such as DroneDeploy, Solo [1]. In orbit mode, a center or Region of Interest (ROI) and a radius are set. After confirming the flight plan, it fixes the camera on the drone towards the ROI and steers the drone along a circular path equidistant around the ROI at the selected radius. There are also softwares, including but not limited to Airnest, Litchi, and Autopilot, that allows the operator to manually select waypoints to generate a spline path around the target with different ROI set for each edge of the path. However, this option requires a manual effort to create a flight plan. Sometimes auto flight mode is used to capture nadir images of a wide target, which also requires manual effort in creating the flight plan. An instance of inspecting an infrastructure target using 3DR-Solo is shown in Fig. 1.1.

There are downsides to the methods mentioned above. Orbit mode pivoting over a single ROI, could cause the UAV Camera to miss capturing some parts of the target partially or completely, especially if the shape of the target is not convex as shown in Fig. 1.2. It would also make the observation distance between UAV and the target vary and make the UAV to have improper visibility of the target over the duration of the video. Performing this task manually is time-consuming and will result in the degradation of the operator's performance due to fatigue [2]. Flying manually to

Figure 1.1: An Instance of Infrastructure Inspection using 3DR-Solo

Figure 1.2: Sequence of snapshots from an inspection. Part of the target had no clear visibility towards left bottom, since the UAV is pointing towards the center.

capture a video of the target is also exhausting on the operator and inefficient. Approaches that require manual effort to perform a part of the inspection task is of a hindrance, especially when the time is critical during post-disaster scenarios. Autonomous capabilities that reduce manual effort could lower the strain on the operator, and speed up the inspection.

The downside of not being able to capture the whole target while conducting inspection using the orbit mode led to the motivating problem: "Can the infrastructure inspection be conducted with more target coverage than the orbit mode?". This thesis introduces a novel method for inspecting infrastructure by capturing a smooth video of the target in a non-circular path with the orientation of the camera perpendicular to the path without a 3d-model of either the target or environment. The presented approach is expected to be as fast (or nearly as fast) as a manual operation. and allow a mediocre pilot to capture the video steady and smooth to perform the inspections effectively, i.e. easily while getting better and more target coverage. The presented approach is also expected to be

helpful in conducting regular inspections of the infrastructure, to keep a track of the conditions and assess the requirement for any damage repairs, especially with a D+ grade for America's overall Infrastructure condition and performance - indicating that the majority of the infrastructure is of serious concern with a strong risk of failure.

This thesis also introduces a methodology and compendium of metrics to quantitatively evaluate and compare varied forms of autonomous infrastructure inspection. This work presents the first formal study of evaluating two modes of autonomous infrastructure inspection. The presented approach is compared against the Orbit mode that allows a smooth video to be captured on a circular path around the target with the camera‘s orientation always perpendicular to the path (pointed to the center).

The remainder of the thesis is organized as follows. Chapter 2 reports the related work and the implementation differences in the presented approach. Chapter 3 describes the approach and Chapter 4 over implementation, which includes the assumptions taken and the work-flow. Chapter 5 goes over the experiments including metrics, and Chapter 6 covers the results, analysis, and real-world demos. Finally, Chapter 7 will conclude and present some future work for this thesis.

2. LITERATURE REVIEW

UAVs have the potential for autonomous defect and damage detection in civil infrastructure, with accuracies that can lead to assessments far more quantitative compared to human inspections [3]. There are works that looked at improving the control scheme of the UAV to allow an unskilled operator easy control and make sure the UAV faces the target [4] [5]. Instead of depending on the operator to manually control the UAV the presented approach focuses on the autonomous approach for inspection. Although there are works that use sensors for obstacle avoidance during waypoint navigation [6], here it is assumed for simplicity that the generated path has no obstacles and instead relies on the operator to takeover the UAV when it is on a collision course.

Among the works that navigate the UAV in predefined flight patterns, Eschmann et. al. controlled a rotorcraft UAV with a high-definition optical camera manually to capture data of a given target in predefined flight patterns and generate a stitched 2d and 3d target reconstructions [7]. Target (bridge) specific flight pattern were used for conducting inspection by Eschmann et. al. [8]. Rather than using predefined flight patterns, that may not work effectively for all scenarios, the presented approach focuses generating a spline path around the target.

Among the works that look at generating trajectories, Mansouri et. al. used a 3d-model of the target to generate collision-free trajectories for cooperative UAVs to inspect the structure [9]. Papachristos et. al. discussed distributed infrastructure inspection by generating a path that was iteratively improved over time constraints and coverage of the target using the traveling salesman problem [10]. Bircher et. al. discussed a real-time capable exploration path planner with boundaries of the area as input to generate the next best-view based on exploration gain over the geometric tree of possible future poses [11]. The presented approach collects the locations during the image capturing phase and generate a spline path for a single UAV to navigate around the target while maintaining the heading perpendicular to the path. This removes dependencies on either a 3d model (and also apriori generation of a 3d model for a given target) or manually marking the area of interest for generating the path.

In order to judge the defect of a target it would be desirable if it is in the center of the image during inspection [12]. The orientation of the camera was fixed in [7] [8] and the paths were preplanned such that the camera faces the target orthogonally to the target. From the preplanned paths it can be observed that the orientation of the camera was also perpendicular to the path. The orientation of the camera was considered during the path generation in [9] [10] [11]. The presented approach focuses on adjusting the camera's orientation perpendicular to the path similar to [7] [8], except instead of preplanned paths, target specific paths are utilized.

No standard set of evaluation method were used across the works discussed above. Since the main intention was to generate a 3d model of the target, no standard set of metrics were used either. Time taken was one of the metric common across the works discussed above. A methodology is introduced in this thesis to compare varied form of autonomous infrastructure inspection by means of capturing a video of the target. This thesis also introduces a set of metrics to quantitatively evaluate and compare the inspection process. More details regarding the evaluation method and metrics are discussed in Chapter 5.

3. APPROACH

The approach is to use the collect locations during the image capture phase and complete the remaining task autonomously by generating a spline path around the target for capturing video and locations for taking nadir from the top of the target. A spline path was chosen because it allows a UAV to navigate smoothly without abrupt speed and direction changes which in-turn allows for a smooth video to be recorded that will allow an inspector to assess the condition of the target without getting distracted and disoriented. Although this approach requires manual effort for image capture phase, it removes manual effort in generating flight plans or executing the rest of the task.

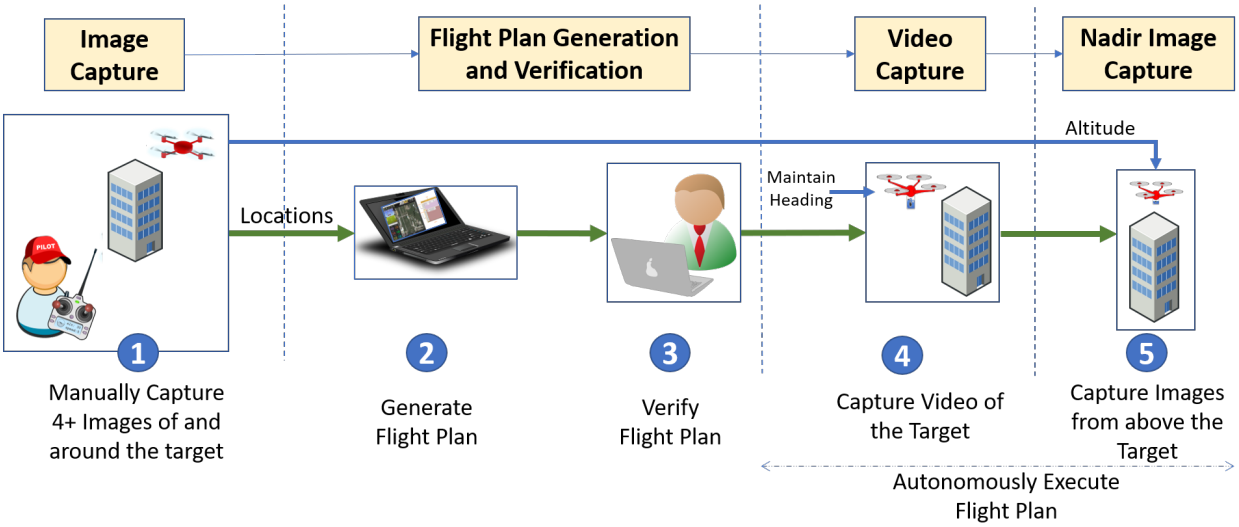

Figure 3.1: Flowchart representation of the presented approach

A flowchart representation of the presented approach outlining the steps to conduct infrastructure inspection is shown in Fig. 3.1. The steps of the approach are as follows,

1. UAV is manually navigated above, away and along the edges of the target, to capture elevation images until the whole of the target is covered. This is generally left to the operator's intuition and expertise to capture relevant viewpoints of the target. While capturing each

image, the location (latitude, longitude, and altitude) of the UAV at which the image was captured will also be stored. More than 3 images are needed to be captured around the target since at least 4 locations are required to generate a path around the target and for the path segments to not go over the target.

2. The locations stored in the above step are sent to the system for processing. Using the locations, a flight plan consisting of a spline path for capturing a smooth video (so that the inspector doesn't get either distracted or disoriented watching the video) and locations for capturing nadir images is created by the presented algorithm. Implementation specifics are detailed in Chapter 4.

3. An operator then views the generated flight plan and verify if the generated flight plan is as intended and safe. After validation, the operator confirms the system to execute the flight plan. UAV then navigates autonomously according to the flight plan to conduct the rest of the inspection task.

4. While navigating for capturing video, the heading of the UAV is kept perpendicular to the course over ground by instructing the UAV regularly. This is because assuming that the generated path segments are parallel to the edges of the target, then having the orientation of the camera perpendicular to the path would allow to cover the target orthogonally and result in a better view for inspection. Video will be captured at an altitude where images are taken during the first step. This is assuming that the altitude at which the image was taken was ample to have the whole target in view and is safe for the UAV to navigate.

5. After capturing the video, the UAV then navigates to the calculated locations and a set altitude (which is estimated during the Image Capture phase since the altitude parameter depends on the height of each specific target which is identified while executing inspection.) to capture nadir images.

4. IMPLEMENTATION

This chapter first lists the specifics of the libraries used and dependencies. It then lists the assumptions made for an algorithmic implementation of the presented approach and goes into the implementation specifics of the presented approach.

The presented approach was implemented as an algorithm and as a proof-of-concept for a 3DR Solo since its one of the popular commercial drones available at a reasonable price that allows developer access to program a task. 3DR Solo is also compatible with an open source framework - Dronekit, that is claimed to be robust and powerful by the developer community [13]. This implementation can be employed by any UAV compatible with the dronekit framework, but the idea can be implemented on any rotor-craft UAV with GPS and Camera. Python was used to implement the approach as it's a high-level language that is supported by the 3DR Solo. Softwares Mission Planner [14] and MAVProxy [15] are used for assistance with the execution of the algorithm. Mission Planner is a fully-featured Ground Control Station (GCS) application for UAV's that can be used as a configuration utility or as a dynamic control supplement for the connected autonomous vehicle. MAVProxy extends GCS to be connected to multiple applications by creating multiple data links. A proof of concept of the implementation is executed on a windows laptop but it works on any computer that supports python and have the required softwares (Mission Planner, MAVProxy) & libraries (dronekit, pymavlink, tkinter) installed. The next sections go through assumptions and a detailed step-by-step implementation in using the system to conduct infrastructure inspection.

4.1 Assumptions

The assumptions made in order to implement the presented approach are:

- A safety officer will be present to ensure that the UAV will not run into obstacles, and also to take over in case of any threat. It cannot be guaranteed that the path generated by the algorithm is obstacle free. Although the UAV was guided around the target by an operator for Image capturing phase, it is possible that the path taken (which need not be spline) is

different from the path generated for capturing the video.

- After executing the flight plan, UAV will return to land at the home location. This is done in order to back up the collected data and replace the battery (if needed) before proceeding to inspect the next target.

- Each inspection task requires an initialization of a new implementation instance. This is done in order to reset the system, initiate the inspection process from the beginning and also to store the logs related to each inspection task separately.

4.2 Image Capture Phase

The UAV is turned on and connected to the developed algorithmic instance and the Mission Planner using MAVProxy in order to keep track of the inspection task and the UAV status. The UAV is taken-off (either manually or autonomous) and then it is manually navigated towards the target. The UAV is then placed above and away from the target, facing the target perpendicularly and approximately bisecting an edge of the target. An image is captured and along with it the location (containing the latitude, longitude, and altitude) of the UAV will be stored by the algorithmic instance. Aerial view of an instance of the UAV capturing an image of the target during the Image Capture phase from the real-world proof of concept trial is shown in Fig. 4.1. The captured image of the target is shown in Fig. 4.2.

This process of capturing images is repeated until the whole of the target is covered and the location of the UAV for each image is stored by the algorithmic instance. Generally, 4+ images are captured for a given target and the choice is left to the operator's intuition and expertise to capture relevant viewpoints of the target. After capturing all the images, the operator communicates to the algorithmic instance that the image capture phase is completed. The operator also provides the parameter value of altitude at which nadir images are to be captured since the value is dependent on the height of the target which is identified while capturing images. The algorithmic instance then uses the locations stored to generate a flight plan for the rest of the inspection task.

Figure 4.1: Capturing an image of the Target using a 3DR-Solo during the Image Capture phase.

Figure 4.2: Captured image of the Target during the Image Capture phase.

4.3 Flight Plan Generation

A flight plan is generated using the locations stored during the image capture phase to autonomously execute the rest of the inspection task i.e, capture a video of the target and capture the

nadir views of the target.

In order to generate a flight plan for the video capture phase, if the collected locations are not in sequence (either clockwise or anti-clockwise), then they are sorted to be in sequence. This is because the generated spline path will be twisty if the locations are not in sequence and result in an ineffective execution of the inspection task. The ordered locations (latitude, longitude, and altitude) of the elevation images are used as waypoints to generate a spline path using the library ArduCopter. The functionality of the spline path in the ArduCopter was made robust by iteratively testing and improving to support windy conditions and high-speed navigation. It compensates any error during the flight by tracking the UAV in real-time and adjusting its speed and orientation, continuously and smoothly, to minimize the error with respect to the generated flight plan. Each segment of the generated spline path is a cubic polynomial curve defined in 3D space with continuous first and second derivatives at the waypoints [16]. This updates calculations faster and allows for a smooth and precise yaw control. This also generates a smooth transition of the UAV's position along and between the curve segments during the flight. It is claimed that time and battery are saved since abrupt speed and direction changes are avoided [16].

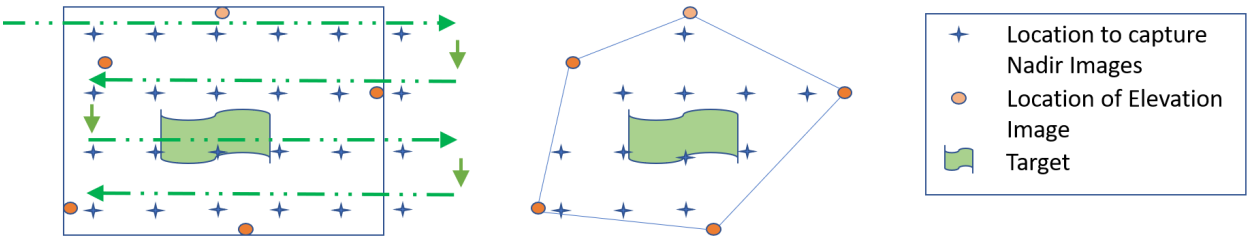

Figure 4.3: Generating locations to capture Nadir images. (a) Generating locations in the bounding box of elevation images locations. (b) Resultant locations for capturing multiple nadir images after removing locations that are outside the convex hull of elevation images locations.

For Nadir image capturing phase, a bounding box for the elevation images locations is used to

generate locations for capturing nadir images. An edge of the bounding box is taken as a bearing reference and locations are generated starting from one corner to the other corner along the selected edge and then back in a zig-zag manner. This process is continued until the whole bounding box is covered, as shown in Fig. 4.3(a). The spacing between locations is predetermined for the UAV camera's footprint and adjusted in proportional to the UAV's altitude set for the nadir images for a given target. After generating all the possible locations, points outside the convex hull polygon of elevation images locations are removed, as shown in Fig. 4.3(b). The altitude for all the locations calculated will be set to the parameter value provided in the image capture phase.

The generated flight plans for capturing video and nadir images are combined and is uploaded to the UAV. The time complexity of generating the combined flight plan is $\mathcal{O}(n \log n)$. The fight plan for capturing video is placed first in the combined flight plan because the UAV will already be at a location (unless it is steered to a different location after capturing the final image) required for capturing video. Since the locations at which the images are captured are used to generate a flight plan for capturing video, the UAV will not have to travel to initiate the video capture phase. The flight plan for capturing nadir images is placed next. A command to do Return-To-Home (RTH) is placed next in the combined flight plan so that the UAV would return and land at the home location.

4.4 Flight Plan Verification

In order to view the generated flight plan, it is first uploaded to the UAV. The operator then uses the Mission Planner to read the flight plan from the UAV as shown in Fig. 4.4, to verify if the generated flight plan is as intended and safe. After verifying the flight plan and classifying it as safe and intended, the operator then confirms the algorithmic instance to proceed with the execution of the flight plan. If the operator identifies a problem with the flight plan, then the inspection task is halted and the process is restarted from the beginning in such a way to rectify the observed problems.

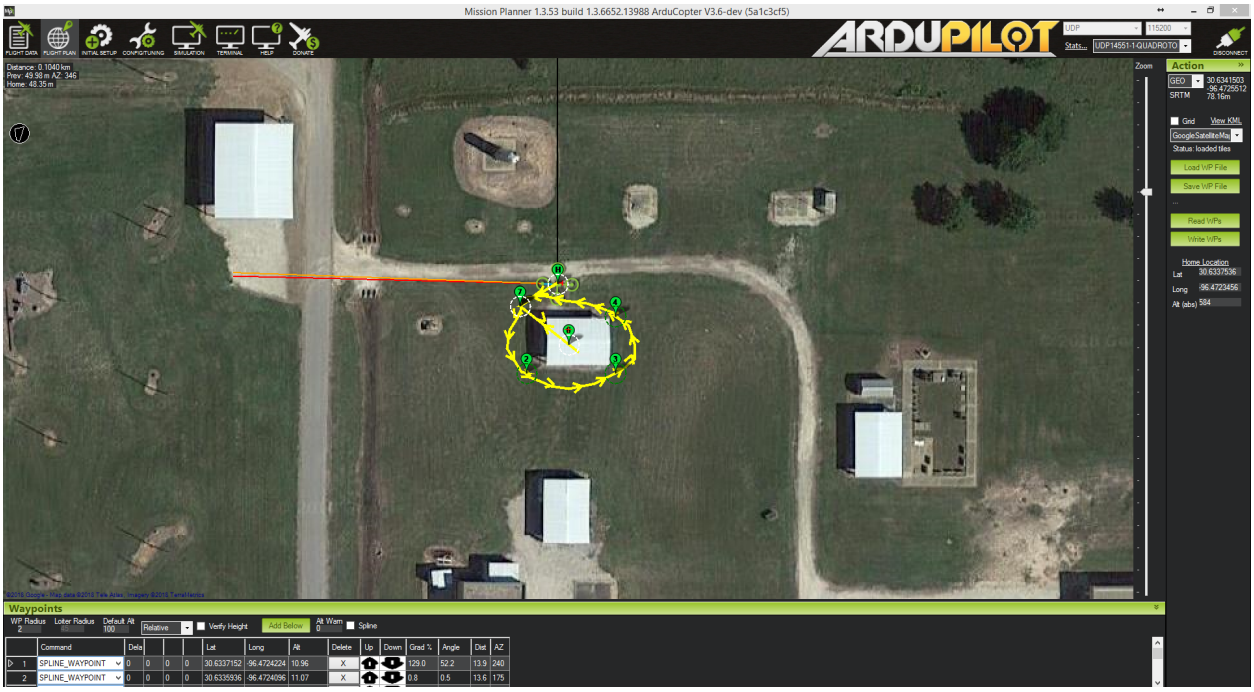

Figure 4.4: Viewing the generated flight plan using Mission Planner to verify.

4.5 Flight Plan Execution

After receiving a confirmation from the operator to proceed with the flight plan, the UAV then autonomously completes the rest of the inspection task. If at any instance the UAV is observed as malfunctioning to either the safety officer or the operator, the control over the UAV can be regained by terminating the algorithmic instance.

In executing the flight plan, the UAV navigates to the first location in the flight plan for recording video and then initiates the camera to start capturing video. UAV then navigates around the building covering all the locations in the flight plan for capturing a smooth video on a spline path. UAV is autonomously commanded by the algorithmic instance regularly, to maintain the heading of its camera perpendicular to the course over ground. When the UAV reaches the final location of the flight plan for recording video, it communicates to the camera to stop capturing video. After terminating to capture video, UAV proceeds with executing the flight plan for capturing nadir images.

In order to capture nadir images, the UAV first commands its camera to point downwards and then navigates to the first location among the calculated locations for capturing nadir images. UAV then commands the camera to capture an image. After capturing the image, UAV navigates to the next location (if available) in the flight plan and continue capturing the nadir images at all the calculated locations. After capturing all such nadir images, the UAV does RTH and lands at the home location.

5. EXPERIMENTS

The main contribution of the presented algorithm is to capture a smooth video covering the target autonomously and result to an effective inspection. The presented algorithm is evaluated against orbit mode for the video capture phase, this chapter first describes the simulation experiments conducted. It then goes into the metrics used to evaluate the performance followed the experiments conducted to demonstrate the real-world proof-of-concept.

5.1 Simulation Experiments

In order to evaluate the presented algorithm, it was compared against orbit mode in simulation using data of 28 targets. These 28 are all the data sets which had nadir view and 4+ elevation images among the data set collected post Hurricane Irma by experts and field agents [17]. The perimeter of each target was identified manually by spotting the corner points of itself in Google Maps and it was assumed that it represents the target accurately. Locations of the elevation images for each target are extracted from the geo-tagged images (sample images for a target are shown in Fig. 5.1). It was assumed here that all the elevation images are captured at the same altitude for the target, and the value was ample to have the target in the view from top to bottom.

For each target, a simulator instance was created at a random perimeter point as home location using dronekit-sitl [18]. The locations extracted from the Geo-tagged images captured during the image capture phase were fed to the algorithmic instance bypassing the manual effort in the creation of waypoints. Using the locations, a flight plan was generated for the target and simulation experiments for the target inspection using the presented algorithm were run for all the targets. For each run, the UAV was taken off at the home location with its speed capped to 2 m/s (default safe cruise speed set by the manufacturer for 3DR Solo) for the flight duration. After confirming the generated flight plan, the UAV autonomously simulated executing the video capture phase and nadir image capture phase. Nadir images were simulated to be captured at an altitude of 30 m, although this could vary up to 120 m in real-world scenarios. The orbit mode trial was simulated

Figure 5.1: Elevation Images of a Target from the dataset

with the center of the target (calculated from the perimeter points above) as the center and the radius equal to the distance between the center of the target and farthest elevation image point. Status of the UAV (including location, altitude, attitude, gimbal orientation, and speed) was logged by the algorithmic instance at a rate of 4 times/sec for metric calculations and analysis.

For the sake of simplicity in the categorization of targets in this work, the interpretation of the convex and concave target was modified and relaxed. Passageways of low separation distance (less than 15 m [19]) were ignored since it is not safe for the UAV to enter it. Upon ignoring all such passageways if the target couldn't be covered continuously without a convex path then the target was tagged as concave else it was tagged as convex. For example, the target shown in Fig. 5.1 was tagged as convex.

5.2 Metrics

Among the works discussed in Chapter 2, the main idea of executing inspection was to generate a 3d-model of the target. The time taken to execute a task was the main quantitative metric used in [11], [9], [20], [21]. This work focuses on inspecting the target by a means of recording a video of the target, and metrics are proposed below to measure its effectiveness. In order to evaluate the inspection process using the presented algorithm against the orbit mode, the following metrics were used. These metrics evaluate only the video capture phase of the inspection task since its the only phase being compared.

5.2.1 Effectiveness w.r.t Target Coverage

To measure the effectiveness of the presented algorithm three metrics are used: Perimeter coverage, Average deviation from perpendicular observation, and Average observation distance. These metrics measure the effectiveness of an algorithm for inspection in terms of target coverage. Covering the whole target with an angle of depression close to 90° and with low observation distance (and low variance) for all the instances is the ideal effective outcome.

5.2.1.1 Perimeter Coverage

Target Coverage is the dominant factor on inspection effectiveness [22]. To express it quantitatively, this metric indicates the percentage perimeter of the target that was covered/observed by the UAV camera during the video capture phase. In order to calculate this metric, the perimeter points identified manually for each target were used along with the locations and orientations of the UAV camera logged for the duration of the video capture phase. Each edge of the target was divided in to sub-edges of length which were dependent on the observation distance. Given that the Field of View (FOV) is 60° for the camera on a 3DR-Solo, the sub-edge length was limited to a maximum of 1/6th of the observation distance so that at any instance if the UAV's camera was facing the target then at least one sub-edge was observed completely. For each log of the UAV's location and its camera orientation if there was a sub-edge in the direct line of sight of the camera, then the sub-edge was marked as covered. After updating the status of each sub-edge for all the

instances logged during the video capture phase, the length of covered sub-edges was counted. The value of this metric was then calculated by dividing the length of all covered sub-edges with the total perimeter for the given target. The mathematical formula for calculating the value of this metric is,

$$Perimeter\ coverage = \frac{\sum_{e_i \in subedges(Target)} length(e_i) * covered(e_i)}{\sum_{e_i \in subedges(Target)} length(e_i)} * 100$$

where,

$$\sum_{e_i \in subedges(Target)} length(e_i) = \text{perimeter of the target}$$

and

$$covered(e_i) = \begin{cases} 1 & \text{if subedge } e_i \text{ was covered} \\ 0 & \text{if subedge } e_i \text{ was not covered} \end{cases}$$

5.2.1.2 Average Deviation in Angle of Depression

The angle of depression indicates the angle between the line joining UAV Camera and the target along the orientation of the camera and the intersecting edge of the target as shown in Fig. 5.2. The target surface is said to be visible if the angle of depression lies between 0° and 180° [23], with 90° providing the efficient view [12]. Instead of calculating the effectiveness of a single viewpoint as in [23], this metric indicates the effectiveness of target coverage w.r.t the angle of depression by the camera to the target surface for every instance logged during the video capture phase. To calculate this metric, the perimeter points identified manually for each target were used along with the locations and orientations of the UAV camera logged for the duration of the video capture phase. In order to measure the effectiveness of the camera angle during the video capture, the deviation of the angle of depression from 90° was calculated since observing the target orthogonally gives a proper view of the target to inspect [23]. The deviation was calculated both w.r.t target and w.r.t UAV. In the measure of the deviation in angle of depression w.r.t the target

the angle at which each sub-edge was covered was taken and its deviation from being orthogonal was calculated. If a sub-edge was observed multiple times, then the angle closest to orthogonal was considered. In the measure of the deviation in angle of depression w.r.t UAV the angle at which the UAV was observing the target at each instance was taken and its deviation from being orthogonal was calculated. If the line originating from the UAV's Camera along its orientation doesn't intersect any edge of the target, then that particular instance was ignored and tagged as not valid. This was done in order to calculate the angle of depression only while observing the target. Deviation angle for all the instances logged and for sub-edges during the video capture phase were calculated and averaged to determine the value of this metric.

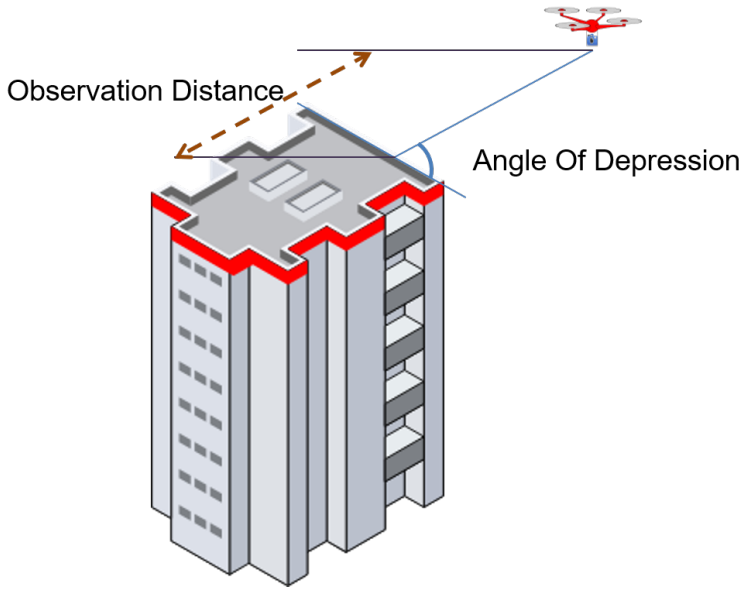

Figure 5.2: Angle of Depression and Observation Distance by the UAV to the target

The mathematical formula for calculating the average deviation in the angle of depression of

the target is,

$$Average\ deviation\ in\ the\ angle\ of\ depression\ of\ the\ target = \frac{\sum_{e_i \in subedges(Target)} covered(e_i) * |angle(e_i) - 90|}{\sum_{e_i \in subedges(Target)} covered(e_i)}$$

where,

$angle(e_i) =$ best angle of depression by the UAV when observing subedge e_i

(here best indicates the instance at which the angle of depression is closest to 90°)

and

$$covered(e_i) = \begin{cases} 1 & \text{if subedge } e_i \text{ was covered} \\ 0 & \text{if subedge } e_i \text{ was not covered} \end{cases}$$

The mathematical formula for calculating the average deviation in the angle of depression by the UAV is,

$$Average\ deviation\ in\ the\ angle\ of\ depression\ by\ the\ UAV = \frac{\sum_{i=1}^{n} observed(i) * |angle\ by\ UAV(i) - 90|}{\sum_{i=1}^{n} observed(i)}$$

where,

$angle\ by\ UAV(i) =$ angle of depression by the UAV on the target at instance i

and

$$observed(i) = \begin{cases} 1 & \text{if UAV had target in view at instance } i \\ 0 & \text{if UAV didn't had target in view at instance } i \end{cases}$$

5.2.1.3 Average Observation Distance

The observation distance indicates the distance between the UAV and the target along the orientation of the camera as shown in Fig. 5.2. A better observation of the target can be made if the observation distance is shorter [23]. To quantitatively compare across all trials as opposed to a single viewpoint in [23], this metric indicates the average distance between the UAV and edge of the target along the orientation of the camera for every instance logged during the video capture phase. In order to calculate this the perimeter points identified manually for each target were used along with the locations and orientations of the UAV camera logged for the duration of the video capture phase. If the line originating from the UAV's Camera along its orientation doesn't intersect any edge of the target, then that particular instance was ignored and tagged as not valid. This was done in order to calculate the observation distance only while observing the target. The observation distance for all valid instances during the video capture phase was calculated, summed, and averaged to determine the value of this metric. The mathematical formula for calculating the average observation distance is,

$$Average\ observation\ distance = \frac{\sum_{i=1}^{n} observed(i) * distance(i)}{\sum_{i=1}^{n} observed(i)}$$

where,

$distance(i) =$ distance between UAV and the target at instance i

along the orientation of the UAV's camera

and

$$observed(i) = \begin{cases} 1 & \text{if UAV had target in view at instance } i \\ 0 & \text{if UAV didn't had target in view at instance } i \end{cases}$$

5.2.2 Cognitive Stability and Comfort

To evaluate the inspection process with respect to cognitive stability and comfort of the operator to inspect a target while watching the video, two metrics were used: Visual Stability and Video Smoothness. If the UAV Camera moves too much during the task then the operator becomes distracted and disoriented. The operator also feels more comfortable if the target was in sight for the whole duration of the video. Hence the inspection process will be better if the UAV camera faces the target for the whole duration and record the video at a low uniform transition.

5.2.2.1 Visual Stability

It is beneficial if the target is in the center of the image for the duration of the inspection [24]. To indicate this quantitatively, this metric indicates the duration for which the camera of the UAV was facing the target during the video capture phase. Any instance that had the target in the line of sight of UAV's camera was marked as valid else it was marked as not valid. To determine the value of this metric the count of valid instances was divided over the total instances for the video capture phase. The mathematical formula for calculating the value of this metric is,

$$Visual\ stability = \frac{\sum_{i=1}^{n} observed(i)}{n} * 100$$

where,

$$observed(i) = \begin{cases} 1 & \text{if UAV had target in view at instance } i \\ 0 & \text{if UAV didn't had target in view at instance } i \end{cases}$$

5.2.2.2 Video Smoothness

An operator favors camera stability more than view quality [25]. To measure the camera stability, this metric indicates the average change in orientation of the UAV's camera over the duration of the video capture phase. In these trials only the yaw was changed for the duration of the video capture phase, pitch and roll are remained fixed. The difference of yaw for each consecutive instances was calculated for all the logged instances during video capture phase. In this work the roll and pitch angles are maintained constant for the duration of the video recording. The summation of all such differences over the count of the instances determined the value of this metric. The mathematical formula for calculating the value of this metric is,

$$Video\ smoothness = \frac{\sum_{i=2}^{n} orientation(i) - orientation(i-1)}{n-1}$$

where,

$orientation(i) =$ absolute yaw angle of the UAV's camera at instance i

5.2.3 Efficiency of the Path

To measure the efficiency of the generated path with respect to the workload on the UAV, two metrics were used: Distance Traveled and Time Taken.

5.2.3.1 Distance Traveled

This metric indicates the distance traveled by the UAV to execute the video capture phase. To determine the value of this metric the location of the UAV for each instance was taken from the log. The distance between two locations of consecutive instances was calculated. Cumulative addition of all such distances for all the instances logged while capturing video determines the value of this

metric. The mathematical formula for calculating the distance traveled by the UAV is,

$$Distance\ traveled = \frac{\sum_{i=2}^{n} distance(location(i),\ location(i-1))}{n-1}$$

where,

$$location(i) = \text{absolute GPS coordinates of the UAV's camera at instance } i$$

and

$$distance(x,\ y) = \text{Approximate distance between two GPS coordinates x, y}$$

5.2.3.2 Time Taken

This metric indicates the time taken for executing the video capture phase. Time stamps were logged when the camera was commanded to start and stop the video recording for the video capture phase. The difference between the time stamps determined the value of this metric. The mathematical formula for calculating the time taken by the UAV is,

$$Time\ taken = (t_{stop} - t_{start})$$

where,

$$t_{start} = \text{time stamp at which video recording was started}$$

and

$$t_{stop} = \text{time stamp at which video recording was stopped}$$

5.3 Real-World Experiments

To demonstrate a proof of concept of the presented algorithm, an experimental inspection was performed using a 3DR-Solo for a target in Veterans Park and Athletic Complex located in College Station, Texas. A Google maps snapshot of the target is shown in Fig. 5.3. This particular target

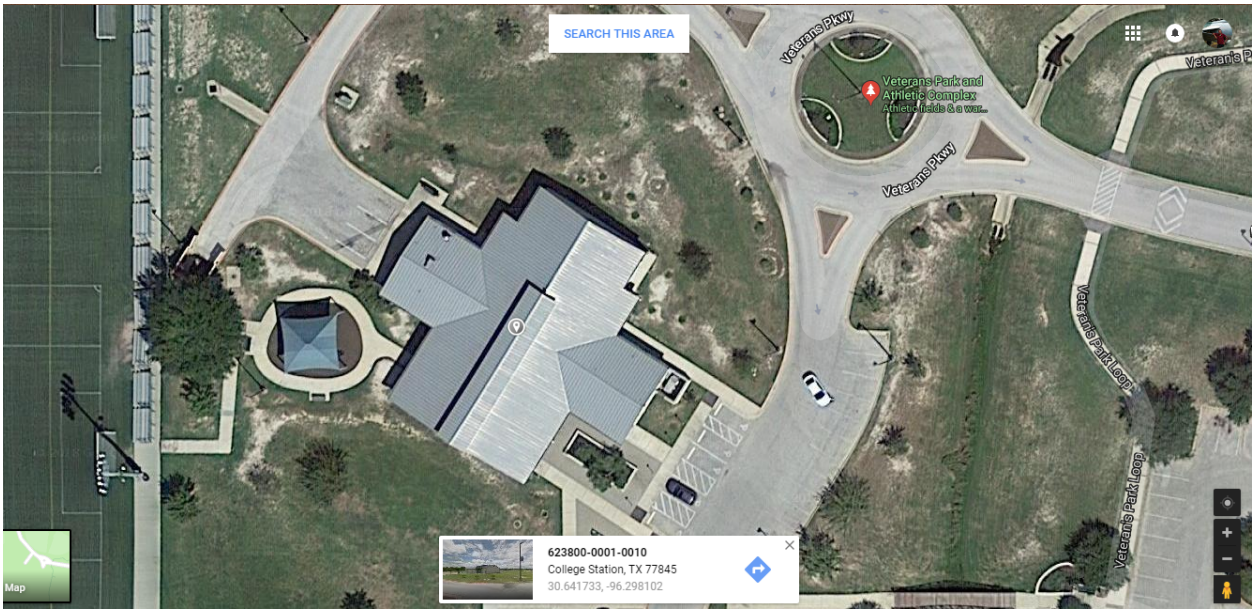

Figure 5.3: Google Maps snapshot of the target for which proof of concept experiment was conducted.

was chosen as it can be treated as both convex and concave target, and to demonstrate the variation in effectiveness of inspection using the presented algorithm and orbit mode. All the trials were conducted under FAA regulations. In order to run a set of experiments with 5 batteries, first, a test flight was conducted to determine the height of the building and the altitude at which the inspection needed to be conducted. Allowing only one inspection per battery, 3 trials were conducted using the presented algorithm and 1 trial was conducted using the orbit mode. Among the trials conducted using the presented algorithm, 1 trial was conducted assuming a convex scenario for the target and other 2 were conducted assuming a concave scenario for the target.

6. RESULTS

This chapter first presents the collected data from the experiments followed by results and analysis from the collected data. From the results it was observed that the presented algorithm is better in terms of perimeter coverage (by covering 6.18% more of the target's perimeter), average angle of depression w.r.t the target (by the target being observed 10.28° closer to orthogonal observation) and average observation distance (by observing the target 18.49 m closer to the target) compared to the orbit mode. Where as the orbit mode is better in terms of visual stability (by having the target in view for 10.53% more duration) and video smoothness (by pivoting the camera 0.3°/frame slower).

6.1 Collected Data

Following the completion of all the simulated trials, the log files generated from each trial and the perimeter of the respective target were used to calculate the metric values listed in Section 5.2. For each metric, an associated results subsection is presented. For easy analysis, all the tables in this section are organized based upon the type of target and the number of waypoints. Targets 1-8 are concave and had more than 4 elevation images. Targets 9-12 are concave but had only 4 elevation images. Finally, Targets 13-28 are convex and had 4 elevation images. The targets were categorized to make an in-depth analysis and evaluate the advantages of both the modes based upon the type of the target.

6.1.1 Perimeter Coverage

The perimeter coverage of the target by the UAV's camera during the video capture phase was calculated by determining the percentage of the target that was covered w.r.t the perimeter of the target. Only the instances that were logged during the video capture were considered. Table 6.1 shows all the collected data from the simulated trials using the presented algorithm and the orbit mode. A positive difference indicates that the perimeter coverage of the target was more using the presented approach. A negative difference indicates that the perimeter coverage of the target was

more using the orbit mode.

6.1.2 Average Deviation in Angle of Depression

The average deviation in the angle of depression was calculated by averaging the angle of depression by the UAV's camera over the target for all instances during the video capture phase. Only the instances where the target was in view of the UAV's camera during the video capture phase were considered since those were determined to be valid for this metric. Table 6.2 shows all the collected data from the simulated trials using the presented algorithm and the orbit mode. The closer the average angle of depression to 90° the better the inspection process as it results in the orthogonal coverage of the target. To show those, the deviation of each angle from 90° was also calculated and included in Table 6.2.

6.1.3 Average Observation Distance

The average observation distance was calculated by averaging the observation distances between the UAV and the target along the orientation of the camera for all instances during the video capture phase. Only the instances where the target was in view of the UAV's camera during the video capture phase were considered since those were determined to be valid for this metric. Table 6.3 shows all the collected data from the simulated trials using the presented algorithm and the orbit mode. This table also shows the standard deviation along with the means for observation distance for an analysis in the variance of the observation distance.

6.1.4 Visual Stability

This metric was calculated by determining the percentage of the instances from the video capture phase that had the target in the camera view. Table 6.4 shows all the collected data from the simulated trials using the presented algorithm and the orbit mode. Higher the value of this metric the better it was for an inspector to inspect a target by watching the captured video without getting distracted or disoriented.

Target ID	Target Covered (%)		Difference (%)
	Algorithm	Orbit	
1	82	75.68	6.32
2	90.35	90.79	-0.44
3	91.19	74.47	16.72
4	80.95	73.17	7.78
5	95.51	84.44	11.07
6	76.53	56.41	20.12
7	93.1	75.64	17.46
8	72.7	52.69	20.01
9	94.29	79.31	14.98
10	84.85	84.34	0.51
11	92.66	88.17	4.49
12	88.24	79.00	9.24
13	89.06	85.19	3.87
14	100	97.30	2.70
15	95.12	100.00	-4.88
16	82.69	75.56	7.13
17	94.57	84.75	9.82
18	93.75	59.38	34.38
19	100	100.00	0.00
20	91.67	100.00	-8.33
21	100	100.00	0.00
22	100	100.00	0.00
23	100	100.00	0.00
24	93.33	100.00	-6.67
25	98.72	100.00	-1.28
26	89.09	90.57	-1.48
27	92.86	96.72	-3.86
28	82.35	76.12	6.23

Table 6.1: Perimeter coverage of the target during the video capture phase using the presented algorithm and the orbit mode from the simulated trials.

Target ID	Avg. deviation in angle of depression from orthogonal observation(°)			
	w.r.t the Target		w.r.t the UAV	
	Algorithm	Orbit	Algorithm	Orbit
1	15.88	28.59	31.22	24.08
2	12.27	36.51	29.97	30.61
3	27.80	38.80	31.19	36.50
4	26.80	26.82	32.09	29.19
5	16.82	23.66	38.47	25.44
6	9.37	32.36	23.17	29.47
7	20.10	33.47	30.38	30.15
8	16.89	32.42	32.82	27.94
9	23.79	43.23	32.09	35.01
10	20.46	47.14	31.65	40.24
11	28.74	46.22	37.29	34.79
12	22.28	37.44	28.32	27.78
13	12.48	30.34	26.08	29.04
14	24.71	29.63	31.60	23.97
15	26.19	35.07	22.90	34.92
16	19.55	23.48	28.72	20.92
17	22.74	19.54	25.75	20.91
18	0.64	4.88	28.52	35.08
19	11.29	0.40	40.70	29.83
20	11.13	23.79	29.36	24.68
21	20.21	25.70	27.19	23.94
22	12.85	3.78	26.06	13.07
23	0.40	22.66	29.38	24.97
24	9.23	19.35	26.10	30.58
25	26.26	36.07	22.95	31.05
26	13.02	28.92	26.75	27.85
27	22.41	33.91	25.36	27.37
28	26.78	24.63	24.92	21.33

Table 6.2: Average deviation from orthogonal observation in angle of depression over the target by the UAV's camera for the video capture phase from the simulated trials using the presented algorithm and the orbit mode.

Target ID	Observation Distance (m)			
	Algorithm		Orbit	
	Mean	Standard Deviation	Mean	Standard Deviation
1	34.56315	13.53477	53.75135	5.323998
2	49.84686	9.512253	80.82285	6.403097
3	22.10246	9.912769	30.86982	9.416429
4	50.88005	15.89882	76.84319	6.33829
5	25.11664	7.601112	46.89745	5.18296
6	53.75817	19.59948	90.49095	13.25984
7	84.54382	15.5859	156.2546	24.8335
8	54.86801	21.49	95.21972	21.46235
9	62.14337	14.43526	76.26776	13.95752
10	35.16174	5.945661	42.08167	7.968317
11	50.15438	13.08619	68.09709	11.1774
12	31.07322	7.316216	42.86205	7.574337
13	28.61698	4.449631	41.958	9.768492
14	49.44442	16.83372	79.54672	5.989092
15	24.45753	9.79051	54.17872	6.811744
16	40.85699	6.635777	50.5583	3.865371
17	21.18876	3.819036	33.31494	9.252321
18	11.06665	1.305663	12.92381	0.820379
19	42.80717	7.722046	56.65009	2.090683
20	17.84003	4.306749	30.94745	3.083257
21	32.84049	4.395825	40.66125	3.597713
22	27.70517	4.701102	40.08898	1.567265
23	30.00903	4.495341	35.70439	3.75949
24	20.19989	2.913207	25.46166	1.1786
25	22.6922	2.635001	48.67304	9.396608
26	27.08268	3.666653	37.27603	7.447811
27	32.53309	4.301935	41.02722	4.730822
28	28.42687	4.697483	40.45867	4.239669

Table 6.3: Observation distance between the UAV and the target along its camera's orientation for the video capture phase from the simulated trials using the presented algorithm and the orbit mode.

Target ID	Visual Stability (%)	
	Algorithm	Orbit
1	81.76	100
2	89.52	100
3	89.15	100
4	73.72	100
5	83.02	100
6	85.65	100
7	91.68	100
8	89.78	100
9	68.46	42.13
10	98.13	100
11	98.38	100
12	98.7	100
13	97.93	100
14	48.27	100
15	80.85	100
16	98.52	100
17	98.62	100
18	95.81	77.78
19	19.5	98.12
20	70.93	100
21	94.63	100
22	73.65	100
23	98.48	100
24	77.89	82
25	95.88	100
26	94.46	100
27	98.34	100
28	97.29	100

Table 6.4: Visual stability values from the simulated trials with the presented algorithm and the orbit mode.

6.1.5 Video Smoothness

This metric was calculated by determining the average shift in the orientation of the camera during the video capture phase. Table 6.5 shows all the collected data from the simulated trials using the presented algorithm and the orbit mode. Lower the value of this metric the better it is for an inspector to inspect a target by watching the captured since the target will move slowly in the video.

6.1.6 Distance Traveled

Distance traveled by the UAV during the video capture phase was calculated by summing the distances between the consecutive instances logged. Only the instances that were logged during the video capture phase were considered. Table 6.6 shows all the collected data from the simulated trials using the presented algorithm and the orbit mode.

6.1.7 Time Taken

Time Taken by the UAV during the video capture phase was determined by calculating the difference between the timestamps of the video capture phase started and stopped. Table 6.7 shows all the collected data from the simulated trials using the presented algorithm and the orbit mode.

6.1.8 Nadir Image Capture phase results

Fig. 6.1 shows the locations generated to capture Nadir images. With the target being wide in Fig. 6.1(a) multiple locations were generated to capture the whole target, whereas only one location was generated for the target in Fig. 6.1(b).

Target ID	Video Smoothness (°/frame)	
	Algorithm	Orbit
1	1.05	0.33
2	0.38	0.22
3	1.04	0.49
4	0.75	0.29
5	0.87	0.43
6	0.77	0.19
7	0.39	0.13
8	0.73	0.15
9	0.42	0.29
10	0.59	0.41
11	0.37	0.25
12	0.52	0.36
13	0.62	0.46
14	0.47	0.29
15	0.74	0.39
16	0.59	0.41
17	0.86	0.54
18	1.91	1.72
19	0.91	0.49
20	1.14	0.75
21	0.69	0.54
22	0.96	0.63
23	0.77	0.62
24	1.39	1.03
25	0.78	0.43
26	0.76	0.54
27	0.6	0.44
28	0.72	0.48

Table 6.5: Video smoothness values from the simulated trials with the presented algorithm and the orbit mode.

Target ID	Distance Traveled (m)	
	Algorithm	Orbit
1	621.71	591.91
2	892.47	886.90
3	357.88	402.87
4	530.55	691.20
5	376.74	449.00
6	922.46	1063.19
7	1362.03	1620.05
8	1147.57	1293.98
9	707.49	690.63
10	517.24	472.45
11	806.08	786.12
12	554.52	546.37
13	415.59	431.49
14	584.66	715.30
15	394.69	505.21
16	488.37	477.08
17	340.35	362.25
18	121.19	112.18
19	417.63	402.28
20	218.41	266.47
21	374.46	364.04
22	290.15	319.93
23	336.66	323.60
24	184.56	192.19
25	355.44	456.82
26	365.42	366.58
27	478.19	440.12
28	410.65	426.48

Table 6.6: Distance traveled by the UAV for the video capture phase from the simulated trials with the presented algorithm and the orbit mode.

Target ID	Time Taken (s)	
	Algorithm	Orbit
1	310	288
2	439	439
3	180	198
4	263	339
5	193	221
6	471	535
7	683	791
8	580	646
9	354	339
10	257	233
11	401	388
12	276	269
13	217	213
14	292	351
15	193	249
16	248	235
17	176	179
18	68	58
19	220	198
20	119	132
21	196	181
22	151	158
23	175	161
24	100	96
25	185	225
26	193	181
27	237	217
28	213	211

Table 6.7: Time taken by the UAV for the video capture phase from the simulated trials with the presented algorithm and the orbit mode.

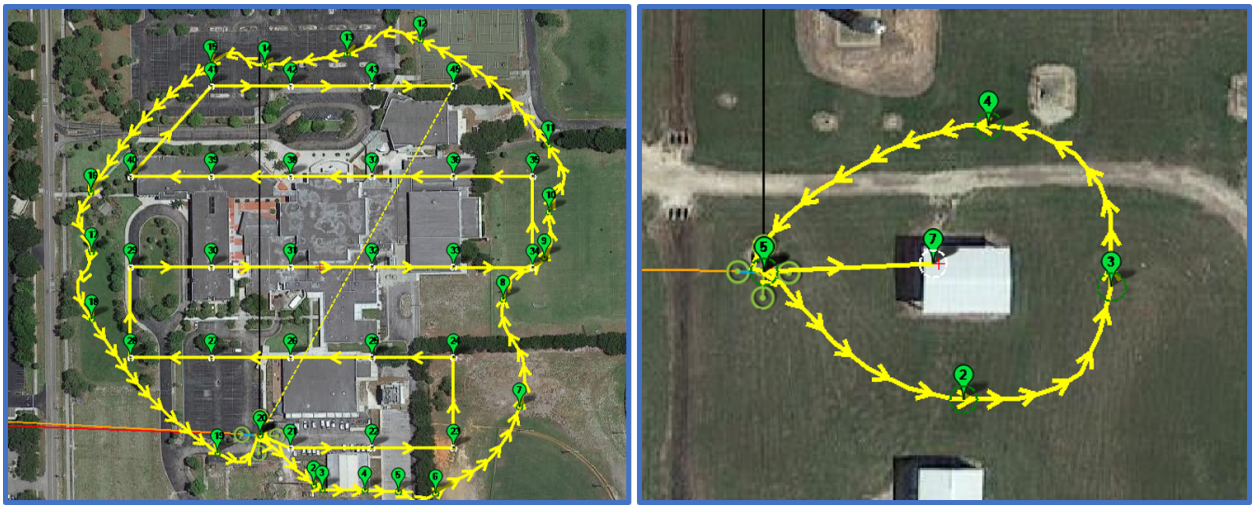

Figure 6.1: (a) Multiple Locations were generated with in the boundary of elevation images locations to capture nadir images. (b) One location was generated to capture nadir image of the whole target.

6.2 Analysis

This section presents the statistical analysis of each metric was performed and evaluated for the presented algorithm and the orbit mode in order to determine which method was better. Plots were also plotted for each target to visually convey the target's perimeter coverage, the path taken by the UAV and the locations of the images captured during the image capture phase.

6.2.1 Perimeter Coverage

Following the completion of all the target's perimeter coverage calculations, the mean and standard deviation for each of the executed mode was calculated. As can be seen in Table 6.8, the target's perimeter was 6.18% more covered on average by using the presented algorithm in comparison to the orbit mode. A statistical two-tailed paired t-test was conducted to determine if the target's perimeter coverage would always be more using the presented algorithm when compared to the orbit mode. A Hypothesis was assumed that there was no contrast between both the modes and the results were shown in Table 6.8. It can be seen that a p-value of 0.002 was found which is less than $\alpha = 0.05$ leading to successfully reject the null hypothesis. A confidence interval was calculated to determine the improvement in using the presented algorithm. It can be said that

for those 28 trials, the target's perimeter coverage was more using the presented algorithm when compared to the orbit mode by a 95 % confidence interval (2.34, 10.03)%.

	Algorithm	Orbit
Mean	91.17 %	84.99 %
Standard Deviation	7.36 %	14.06 %
t Stat	3.29294	
P(T $\leq$ t) two-tail	0.00277	
t Critical two-tail	2.770683	
95% Confidence	(2.34, 10.03) %	

Table 6.8: Summary of target's perimeter coverage.

6.2.2 Average deviation in Angle of Depression

Following the completion of all the average deviation in angle of depression calculations, the mean and standard deviation for each of the executed mode was calculated. As can be seen in Table 6.9, the average of average angle of depression w.r.t the target was 10.28° closer to 90° using the presented algorithm in comparison to the orbit mode. It can also be seen from Table 6.10 that the average angle of depression w.r.t the UAV was 1.08° away from 90° using the presented algorithm in comparison to the orbit mode. A statistical two-tailed paired t-test was conducted to determine if the average angle of depression would always be closer to 90° using the presented algorithm when compared to the orbit mode. A Hypothesis was assumed that there was no contrast between both the modes and the results were shown in Table 6.9 and Table 6.10. It can be seen that a p-value of 4.95 x 10⁎06 was found which is less than α = 0.05 leading to successfully reject the null hypothesis for deviation in angle of depression w.r.t the target. A confidence interval was calculated to determine the improvement in using the presented algorithm. It can be said that for those 28 trials, the average angle of depression w.r.t the target was closer to 90° when compared to the orbit mode by a 95 % confidence interval (6.56°, 13.99°). It can be also seen that a p-value

of 0.39 was found which is more than α = 0.05 failing to reject the null hypothesis for deviation in angle of depression w.r.t the UAV. In conclusion, it can be said that for those 28 trials, neither the presented algorithm nor the orbit mode resulted in statically more average orthogonal observation by the UAV.

	Algorithm	Orbit
Mean	17.89°	28.17°
Standard Deviation	7.78°	11.50°
t Stat	-5.68	
P(T $\leq$ t) two-tail	4.95E-06	
t Critical two-tail	2.051830516	
95% Confidence	(-6.56°, -13.99°)	

Table 6.9: Summary of deviation in average angle of depression from orthogonal w.r.t the target.

	Algorithm	Orbit
Mean	29.32°	28.24°
Standard Deviation	4.45°	5.74°
t Stat	0.87	
P(T $\leq$ t) two-tail	0.39	
t Critical two-tail	2.051830516	
95% Confidence	-	

Table 6.10: Summary of deviation in average angle of depression from orthogonal w.r.t the UAV.

6.2.3 Average Observation Distance

Following the completion of all the average observation distance calculations, the mean and standard deviation for each of the executed mode was calculated. As can be seen in Table 6.11, the

average of average observation distance was 18.49 m less using the presented algorithm in comparison to the orbit mode. A statistical two-tailed paired t-test was conducted to determine if the average observation distance would always be less using the presented algorithm when compared to the orbit mode. A Hypothesis was assumed that there was no contrast between both the modes and the results were shown in Table 6.11. It can be seen that a p-value of 2.88 x 10^{-7} was found which is less than α = 0.05 leading to successfully reject the null hypothesis. A confidence interval was calculated to determine the improvement in using the presented algorithm. It can be said that for those 28 trials, the average observation distance was less using the presented algorithm when compared to the orbit mode by a 95 % confidence interval (12.89, 24.11) m.

	Algorithm	Orbit
Mean	36.14 m	54.63 m
Standard Deviation	28.48	16.03
t Stat	6.767	
P(T $\leq$ t) two-tail	2.88E-07	
t Critical two-tail	2.770683	
95% Confidence	(-12.89, -24.11) m	

Table 6.11: Summary of average observation distance metric.

6.2.4 Visual Stability

Following the completion of all the visual stability calculations, the mean and standard deviation for each of the executed mode was calculated. As can be seen in Table 6.12, the presented algorithm was on average 10.53% visually less stable in comparison to the orbit mode. A statistical two-tailed paired t-test was conducted to determine if the presented algorithm would always be visually less stable when compared to the orbit mode. A Hypothesis was assumed that there was no contrast between both the modes and the results were shown in Table 6.12. It can be seen that a p-value of 0.0042 was found which is less than α = 0.05 leading to successfully reject the null

hypothesis. A confidence interval was calculated to determine the variation in using the presented algorithm. It can be said that for those 28 trials, the inspection was visually less stable using the presented algorithm when compared to the orbit mode by a 95 % confidence interval (3.59, 17.48) %.

	Algorithm	Orbit
Mean	86.12 %	96.66 %
Standard Deviation	17.2	11.3
t Stat	-3.09874	
P(T $\leq$ t) two-tail	0.004198	
t Critical two-tail	2.042272	
95% Confidence	(-3.59, -17.48) %	

Table 6.12: Summary of visual stability metric.

6.2.5 Video Smoothness

Following the completion of all the video smoothness calculations, the mean and standard deviation for each of the executed mode was calculated. As can be seen in Table 6.13, the presented algorithm was on average 0.3°/frame faster in pivoting when compared to the orbit mode. A statistical two-tailed paired t-test was conducted to determine if the presented algorithm would always pivot less smoothly when compared to the orbit mode. A Hypothesis was assumed that there was no contrast between both the modes and the results were shown in Table 6.13. It can be seen that a p-value of 2.22 x 10^{-10} was found which is less than α = 0.05 leading to successfully reject the null hypothesis. A confidence interval was calculated to determine the variation in using the presented algorithm. It can be said that for those 28 trials, the UAV pivoted less smoothly using the presented algorithm when compared to the orbit mode by a 95 % confidence interval (0.37, 0.24)°/frame.

	Algorithm	Orbit
Mean	0.78°/frame	0.475°/frame
Standard Deviation	0.33°/frame	0.31°/frame
t Stat	9.793957	
P(T $\leq$ t) two-tail	2.22E-10	
t Critical two-tail	2.051831	
95% Confidence	(0.24, 0.37)°/frame	

Table 6.13: Summary of video smoothness metric.

6.2.6 Distance Traveled

Following the completion of all the distance traveled by the UAV calculations, the mean and standard deviation for each of the executed mode was calculated. As can be seen in Table 6.14, on average the UAV traveled 38.7 m less using the presented algorithm when compared to the orbit mode. A statistical two-tailed paired t-test was conducted to determine if the UAV would always travel less using the presented algorithm when compared to the orbit mode. A Hypothesis was assumed that there was no contrast between both the modes and the results were shown in Table 6.14. It can be seen that a p-value of 0.01 was found which is less than α = 0.05 leading to successfully reject the null hypothesis. A confidence interval was calculated to determine the variation in using the presented algorithm. It can be said that for those 28 trials, the UAV traveled less using the presented algorithm when compared to the orbit mode by a 95 % confidence interval (9.93, 67.46) m.

6.2.7 Time Taken

Following the completion of all the time taken by the UAV calculations, the mean and standard deviation for each of the executed mode was calculated. As can be seen in Table 6.15, on average the UAV completed 12.17 s faster using the presented algorithm when compared to the orbit mode. A statistical two-tailed paired t-test was conducted to determine if the UAV would always complete inspection faster using the presented algorithm when compared to the orbit mode. A Hypothesis

	Algorithm	Orbit
Mean	520.47 m	559.17 m
Standard Deviation	284.69 m	328.99 m
t Stat	2.760289	
P(T $\leq$ t) two-tail	0.01025	
t Critical two-tail	2.051831	
95% Confidence	(-9.93, -67.46) m	

Table 6.14: Summary of distance traveled metric.

was assumed that there was no contrast between both the modes and the results were shown in Table 6.15. It can be seen that a p-value of 0.08 was found which is not less than α = 0.05 and failing to reject the null hypothesis. This means that there was not enough evidence available to suggest the null hypothesis was false at the 95% confidence level. In conclusion, it can be said that for those 28 trials, neither the presented algorithm nor the orbit mode resulted in statically faster inspection.

	Algorithm	Orbit
Mean	263.9 s	276.1 s
Standard Deviation	162.3 s	140.67 s
t Stat	1.801545	
P(T $\leq$ t) two-tail	0.082795	
t Critical two-tail	2.051831	
95% Confidence	-	

Table 6.15: Summary of time taken metric.

6.3 Observations

This section points out the observations made either from the simulated experiments or data analysis. It also highlights the trials that had the highest variations in terms of the metric values

either positively or negatively using the presented algorithm in comparison to the orbit mode.

From the metric values for the perimeter coverage in Table 6.1, it can be seen that the target's perimeter coverage was either approximately same or better for all the concave targets from the simulated inspection trials using the presented algorithm. Only 5 among the convex targets had more target's perimeter coverage using the presented algorithm indicating that the presented algorithm was more beneficial in getting more target's perimeter coverage for concave targets than convex targets. It was also observed to be statistically significant that the presented algorithm performed better in terms of target's perimeter coverage compared to the orbit mode from section. 6.2.1. It was observed that 22 out of 20 targets had more close to perpendicular coverage (target coverage with an angle of depression $\in$ [80°, 100°]) of the target from the simulated inspections using the presented algorithm compared to the orbit mode.

From the metric values for the average deviation in angle of depression in Table 6.2, it can be seen that the angle of depression w.r.t the target was always close to 90° for all the concave targets from the simulated inspection trials using the presented algorithm. 12 among the convex targets had a better average angle of depression w.r.t the target using the presented algorithm. This indicates that the presented algorithm was more beneficial in covering the target closer to orthogonal than the orbit mode. This result was also observed to be statistically significant in section. 6.2.2. Only 12 among all the targets had the UAV observing the target more perpendicularly on average. No statistical significance was observed for the angle of depression w.r.t the UAV.

From the metric values for the average observation distance in Table 6.3, it can be seen that the average observation distance was always less by using the presented algorithm for all the targets. It was also observed to be statistically significant that the presented algorithm captured the target closer compared to the orbit mode from section. 6.2.6 and the target had been captured with a better view because of low observation distance [23]. It was also observed that the standard deviation in the observation distance was more using the presented algorithm by 1 m in comparison with the orbit mode indicating that the presented algorithm could lead to a more improper visibility of the target than the orbit mode.

From the metric values for the visual stability in Table 6.4, it can be seen that the orbit mode always had the target in view with the exception of three cases. The UAV during the inspection simulated using the orbit mode for Target-18 shown in Fig. 6.9 and Target-24, had relatively high transition speed for the radius that the yaw rate couldn't keep up with it. Target-9 had multiple subjects leading to an ROI that was within neither of the subjects. It was also observed to be statistically significant that the presented algorithm was observing the target for a lesser duration and that could lead to the operator getting distracted and disoriented watching the video.

From the metric values for the video smoothness in Table 6.5, it can be seen that the orbit mode always had lower pivot rate with the exception of Target-18 compared to the presented algorithm. It was also observed to be statistically significant that the presented algorithm had high average pivot rate, and that could lead to faster movement of the target in the video. More calculations were made for further observations and it was observed that the max pivot between any two consecutive instances among all trials was 5° (Target-18) for the orbit mode and 34° (Target-5) for the algorithm. There were 10 trials of the presented algorithm that had more than 10° pivot compared to none among the orbit mode trials. This was due to the high curvature paths generated for the presented algorithm trials. It can be seen from Targets 1 and 5 shown in Fig. 6.2 and Fig. 6.5 below, that closer waypoints result in a higher curvature path. This conveys the dependence on the location of waypoints for the generation of the spline path.

From the metric values for the distance traveled in Table 6.6, it can be seen that the distance traveled by the UAV was less in 16 trials compared to orbit mode. But it was also statistically observed that the UAV executed video capture phase by traveling less distance by using the presented algorithm compared to the orbit mode from Table 6.14. From the metric values for the time taken in Table 6.7, it can be seen that the UAV executed the video capture phase faster using the presented algorithm only in 12 cases. No statistical significance was observed for the time taken to execute the video capture phase from Table 6.15.

Target trials that had interesting observations are shown via the plots below. The following plots show the target, its coverage by the UAV, and the path taken by the UAV from the simulated

inspections using the presented algorithm and the orbit mode. Each plot convey the path taken by the UAV colored in Black, with a point on the path in Dark Black if the UAV had the target in view when it was at that position else the point was marked in Grey. The plot for the algorithm trial contains the waypoints in diamond-shaped marker. The target in the plot for both modes shows the coverage colored accordingly to the angle of depression on it with green indicating the angle was close to 90° and red away from 90° (towards either 0° or 180°).

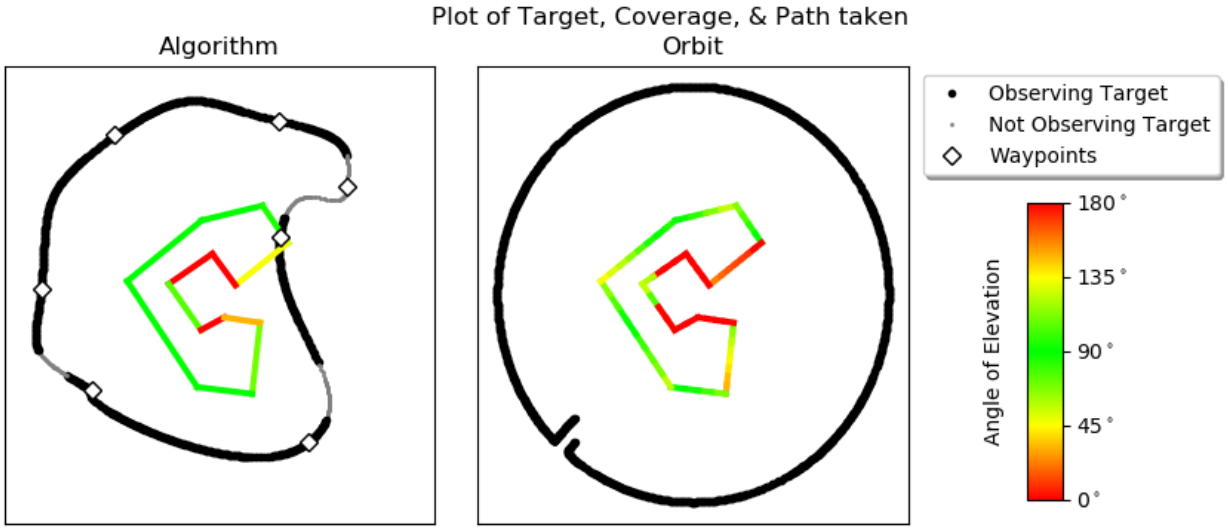

Figure 6.2: Target-1, an example to show the orbit mode not sufficient in terms of target's perimeter coverage for concave targets

Target-1 shown in Fig. 6.2 is a sample trial showing that the orbit mode was not sufficient to get proper coverage of the target for concave targets. It also displays that the performance of the presented algorithm can be disrupted by a waypoint at an odd location.

Target-2 shown in Fig. 6.3, although had no advantage with respect to target's perimeter coverage it can be seen that it had better angle of depression using the presented algorithm compared to the orbit mode.

Target-3 shown in Fig. 6.4 had high advantage in average angle of depression w.r.t the target because of the placement of the waypoints. This target had multiple subjects to be inspected and

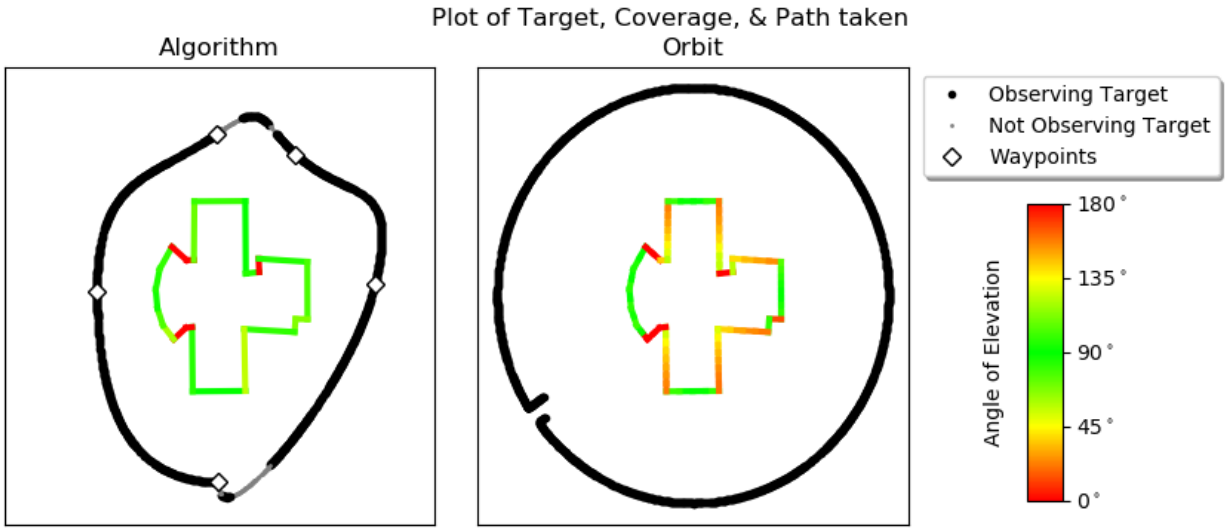

Figure 6.3: Target-2 had the least advantage in target's perimeter coverage among the concave targets

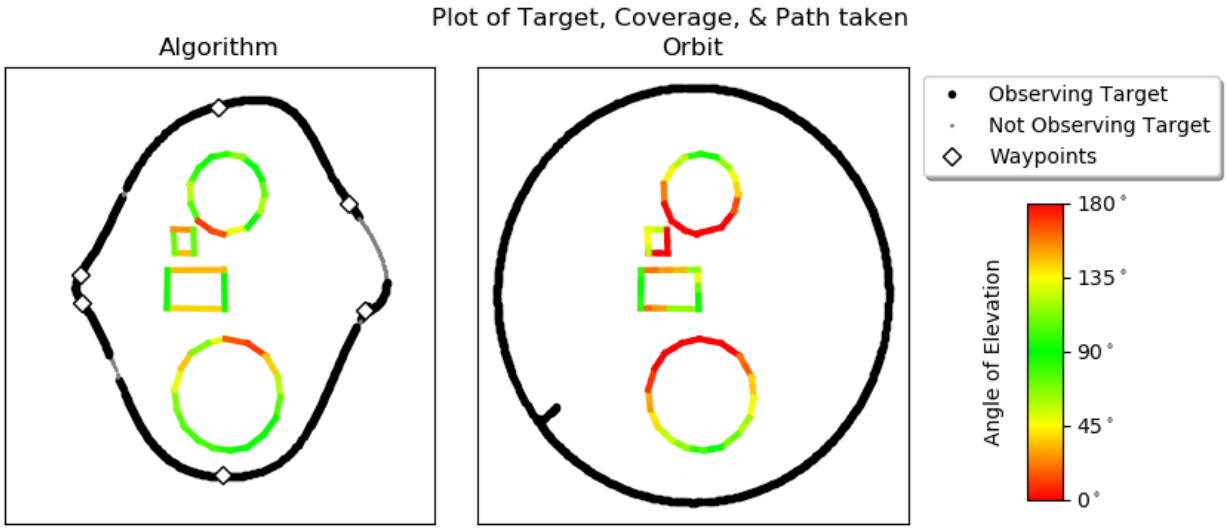

Figure 6.4: Target-3 had high advantage in average angle of depression even with multiple targets

the waypoints helped to inspect them effectively.

Target-5 shown in Fig. 6.5 had high curvature path generated leading to the highest pivot of 34° between consecutive instances.

Target-6 shown in Fig. 6.6, had the highest advantage of target's perimeter coverage among

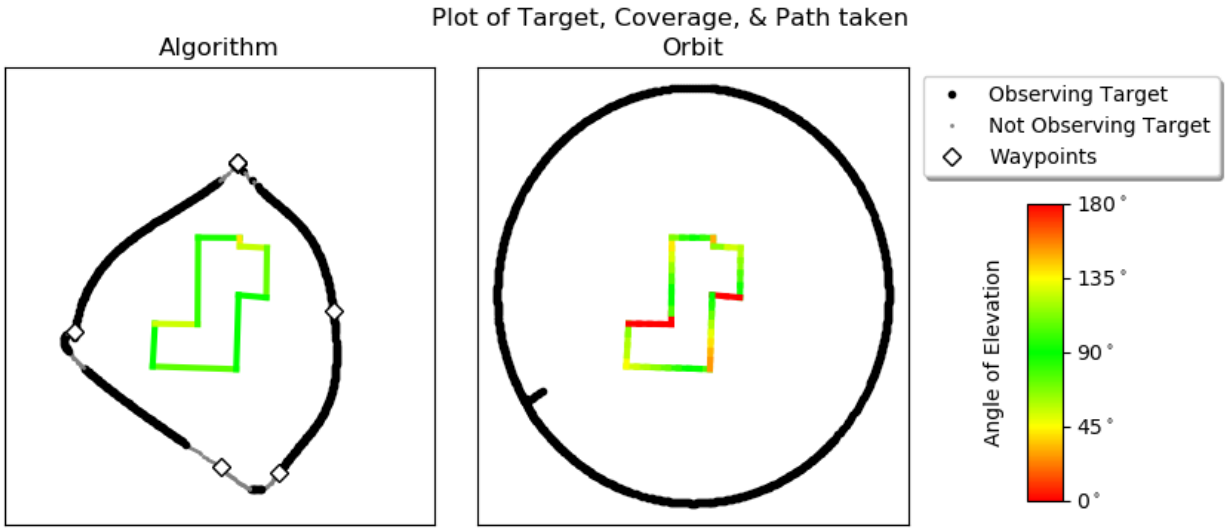

Figure 6.5: Target-5 had high curvature path generated due to waypoints

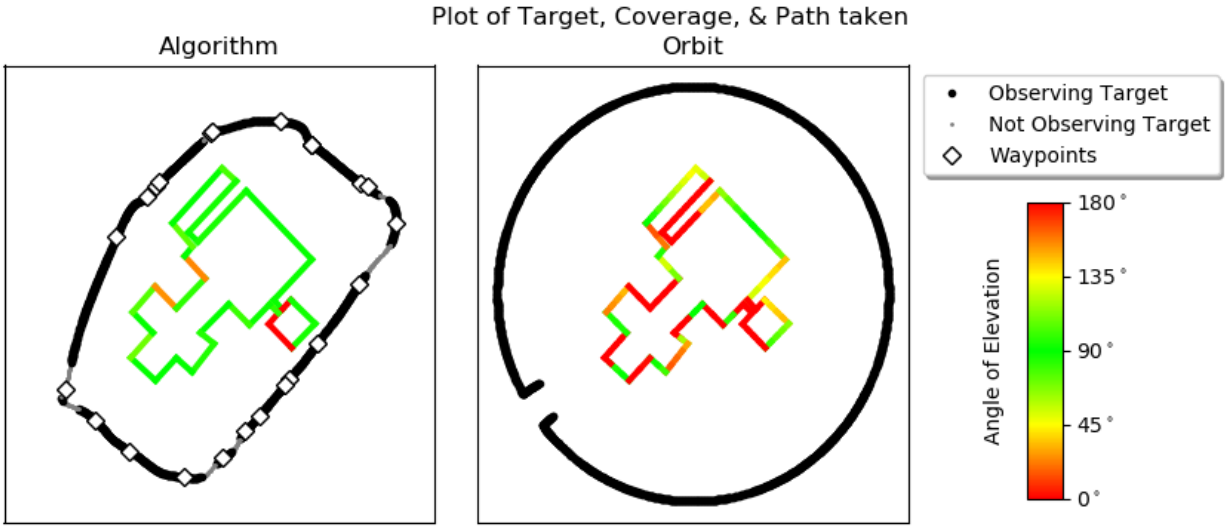

Figure 6.6: Target-6 had the highest advantage in target's perimeter coverage among the concave targets. It also had the highest advantage of close to perpendicular target's perimeter coverage among all the targets

the concave targets by using the presented algorithm because it had many (24) waypoints which helped the UAV to get more target's perimeter coverage as opposed to the orbit mode.

Target-7 shown in Fig. 6.7 had 12 waypoints that helped in generating a spline path using the

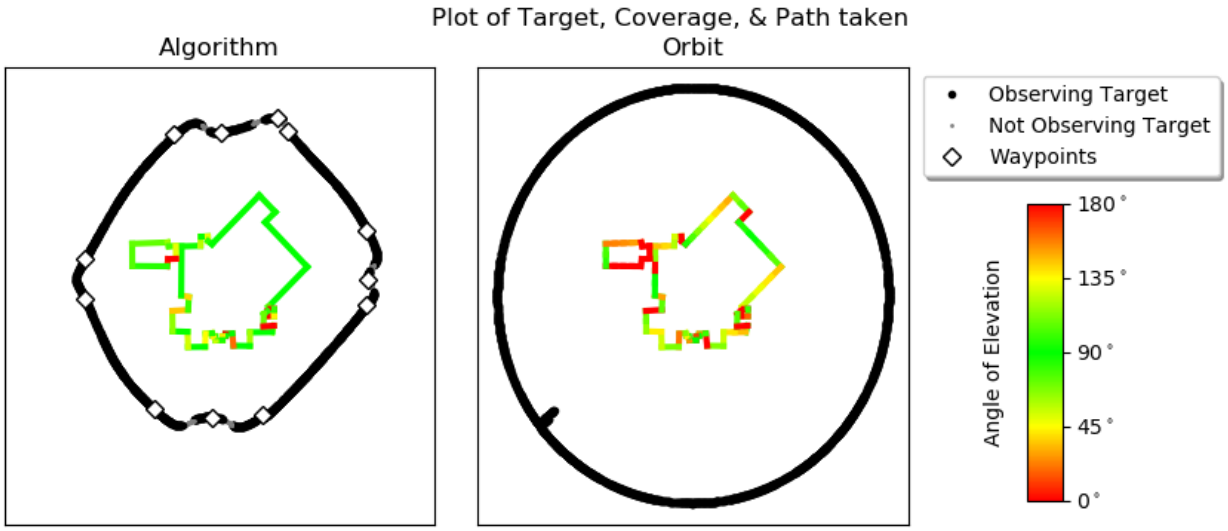

Figure 6.7: Target-7 had relatively low standard deviation in observation distance (15.6m) for the trial with the presented algorithm compared to the orbit mode (24.8m).

presented algorithm, around the target with low variation in the observation distance. Having more waypoints could generate a spline path with low observation distance.

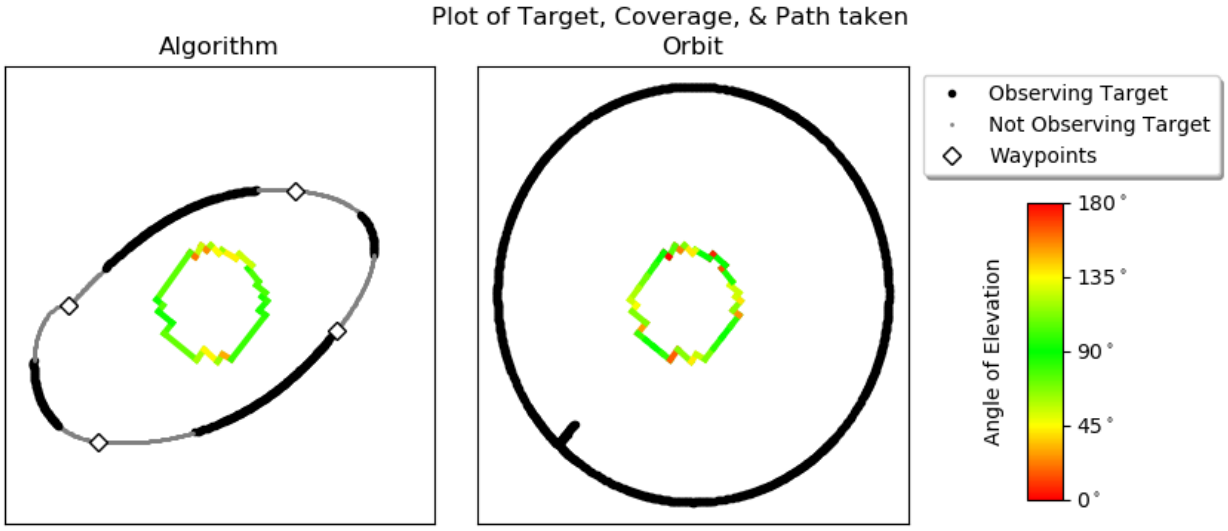

Figure 6.8: Target-14 had highest relatively high standard deviation in observation distance for the trial with the presented algorithm (16.83 m) compared to the orbit mode (5.99 m).

Target-14 shown in Fig. 6.8 had high variation in observation distance for the generated path using the presented algorithm. The locations of the waypoints were improperly aligned with the target that led to the generation of the path with its axis misaligned with the axis of the target. This caused the path edges misaligned with the edges of the target that lead to a high variation in the observation distance.

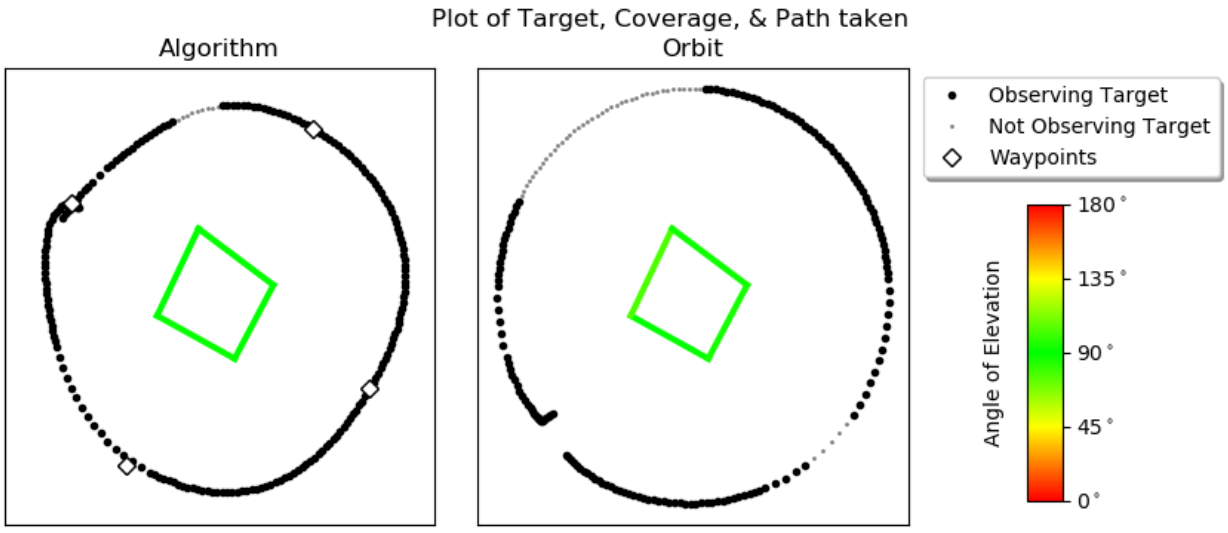

Figure 6.9: Target-18 had the highest advantage in target's perimeter coverage among all the targets.

Target-18 shown in Fig. 6.9 had the highest advantage in target's perimeter coverage among all the targets because of an exception in the execution of the orbit mode. The pivot of the UAV wasn't able to keep up with the transition speed of the UAV because of relatively low observation distance. Generally, the maximum speed in the orbit mode is varied and dependent on the radius set so that the yaw rate of UAV keeps up with the transition speed of the UAV. Since the speed was capped to the same value for all the trials, the speed limit set was high for the trial of Target-18.

Target-20 shown in Fig. 6.10 had relatively less target's perimeter coverage in the presented algorithm trial compared to the orbit mode. This was because of the placement of the location of a waypoint which was relatively farther from the target compared to the rest of the waypoints.

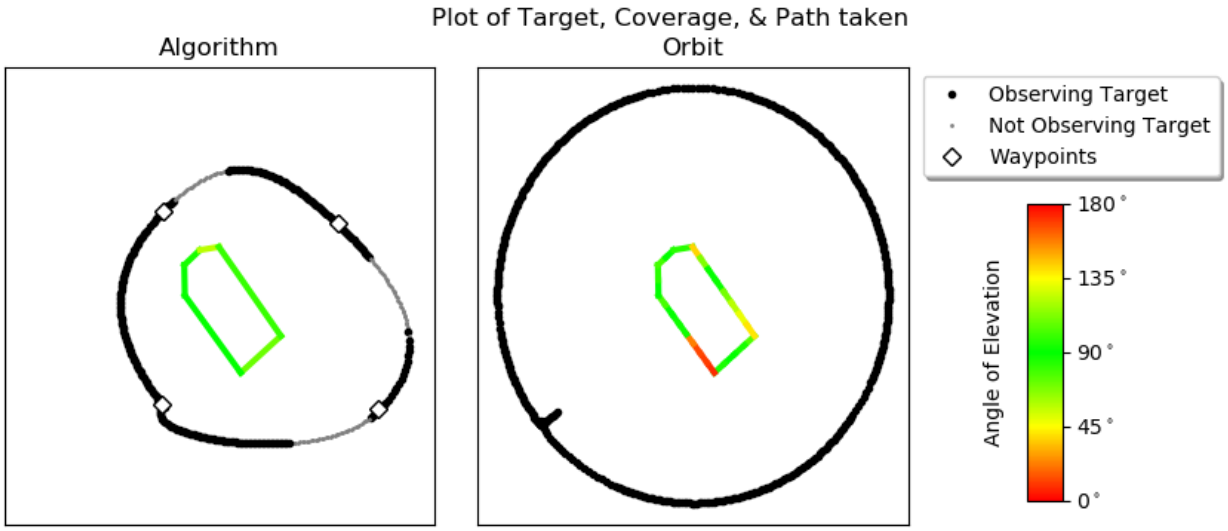

Figure 6.10: Target-20 had relatively the worst advantage in target's perimeter coverage among the targets.

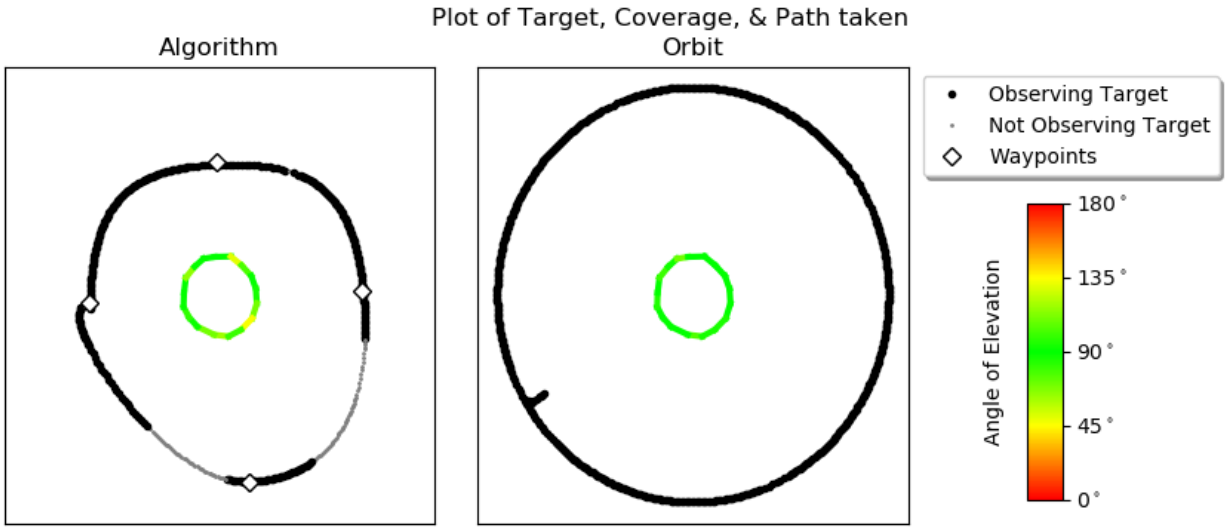

Figure 6.11: Target-22 had the worst advantage in close to perpendicular target's perimeter coverage among the targets.

Target-22 shown in Fig. 6.11 had relatively low perpendicular target's perimeter coverage using the presented algorithm because of the placement of a waypoint away from the desired position and also the shape of the target. With the target being circular in shape (Fig. 6.12) a circular path

Figure 6.12: An elevation image of Target-22.

with the UAV pointed to its center will always cover the target perpendicularly.

Assuming that the dataset represents the variation in targets and expert‘s intuition/identification of locations to capture elevation images, it can be seen that presented algorithm was more effective for concave than convex targets (10% vs 2%, 15.45° vs 5.8°, 25 m vs 13 m, respective improvement for % target covered, avg. angle of depression w.r.t the target and avg. observation distance). Some of the concave targets had more than 4 elevation images being taken, indicating that the target was complex and 4 viewpoints couldn't capture the details of the target. Where as all the convex targets had only 4 elevation images captured. Orbit mode was more effective compared to the presented algorithm for circular and square shaped targets since orbit mode points UAV to a single ROI, i.e the target's center as can be seen in Fig. 6.11 for Target-22. Orbit mode can be pictured as a special case of the presented algorithm with a circular path for capturing the video.

While Orbit depends on an ROI, the presented algorithm depends on the waypoint locations. Orbit can miss the target if the selected ROI was not at the center of the target, whereas the presented algorithm can be visually unstable depending on the spline path generated by the waypoints.

6.4 Real-World Experiments

For the flights conducted to demonstrate a proof-of-concept, first, a test flight was conducted to determine the safe height to inspect the target. There were obstacles around the target of height 28m. It was then determined that the safe altitude to fly without colliding into obstacles was around 32 m. The expertise of a Lead Pilot from CRASAR - Justin Adams, was taken while conducting these flights. For the first flight, the speed of the UAV capped to 2 m/s and four images are captured (along with the locations). Then the algorithmic instance was communicated the end of image capture phase and it generated a spline path around the target for the video capture phase and locations for the nadir image capture phase. After executing the flight plan, it was observed that the speed limit was low and the inspection task barely got completed with 10% battery left. To compensate for this issue, the speed limit was increased to 3 m/s. Then a concave scenario was assumed and then 2 flights of target inspection were conducted (one with 7 locations and one with 8 locations). Then another flight was flown using the orbit mode to capture a video around the target with the speed capped to default setting by the software Solo to 2 m/s.

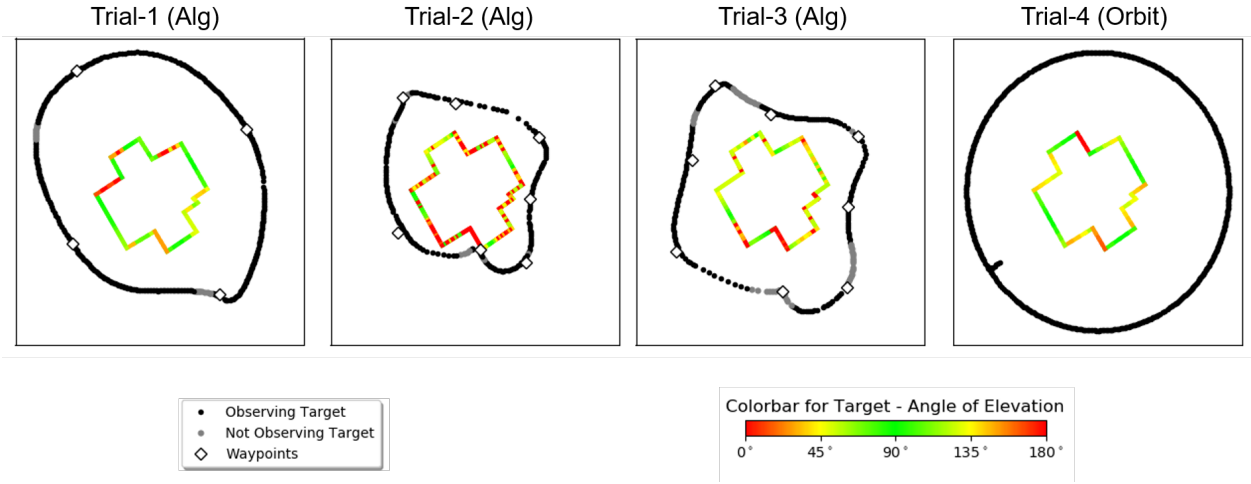

Figure 6.13: Plots for the trials conducted at Veterans Park and Athletic Complex.

The plots for these flight are shown in Fig. 6.13 and the metrics in Table 6.16. From the video

	Trial-1 (Alg)	Trial-2 (Alg)	Trial-3 (Alg)	Trial-4 (Orbit)
Perimeter coverage (%)	93.83	58.43	83.33	94.74
Avg. deviation in angle of depression w.r.t the Target (°)	22.86	21.75	24.80	31.9
Avg. deviation in angle of depression w.r.t the UAV (°)	29.52	35.58	37.05	28.88
Avg. Observation Distance (m)	24.79	11.12	16.96	30.37
Visual Stability (%)	94.74	94.53	76.47	100.00
Video Smoothness (°)	1.25	3.38	2.8	0.61
Distance Traveled (m)	296.10	214.05	263.15	334.47
Time Taken (s)	138	65	80	158
Max. Speed (m/s)	2	3	3	2

Table 6.16: Metric values from the real world trials.

recording it was observed that along with waypoints speed also effects visual stability. Conducting inspection of the target by using the presented algorithm as a convex scenario (Trial-1) was more effective compared to the rest. More orthogonal coverage of the target at low observation distance was observed in the presented algorithm trial with a convex scenario for the target. Inspection with the Orbit mode (Trial-4) was better than the presented algorithm with the concave assumption of the target (Trial - 2, 3) in every aspect, in this particular experiment as shown in Table 6.16.

7. CONCLUSION

This thesis presented and demonstrated a novel idea for autonomous infrastructure inspection using UAVs. The inspection procedure requires the operator to capture images along the edges of the target (4 or more depending on the target) and store locations simultaneously. Using the locations a flight plan consisting of a spline path for video and locations for nadir views is created. While Orbit was better in terms of Visual Stability and Video Smoothness, the presented algorithm was better in terms of Coverage, Average Angle of Depression w.r.t the target, and Average Observation Distance. From the analysis, it had been identified that the performance of the presented algorithm depends a lot on the waypoints. Orbit mode executed the task effectively for simple convex targets, whereas the presented algorithm was effective for concave targets, especially when there are more than 4 waypoints.

More real-world experiments are needed to identify the better values for speed, count and positioning of elevation images, and altitude. The performance of the presented algorithm is speculated to improve by utilizing the orientation of the UAV along with the location during elevation image capturing. Navigating along the spline path while shifting the orientation between the recorded orientations on the waypoints, could make sure that the UAV is facing the target. It can also be improved by exploiting a mix of the spline and normal waypoints to avoid higher curvature paths. Splitting the video capture phase and nadir image capture phase could improve the performance of the presented algorithm. In splitting the missions, the spline path of the video capture phase generation becomes independent of the locations of the nadir image capture phase.

REFERENCES

[1] "Orbit | 3dr." Web.

[2] L. Feng, C. Wiltsche, L. Humphrey, and U. Topcu, "Controller synthesis for autonomous systems interacting with human operators," in *Proceedings of the ACM/IEEE Sixth International Conference on Cyber-Physical Systems*, pp. 70–79, ACM, 2015.

[3] A. Ellenberg, L. Branco, A. Krick, I. Bartoli, and A. Kontsos, "Use of unmanned aerial vehicle for quantitative infrastructure evaluation," *Journal of Infrastructure Systems*, vol. 21, no. 3, p. 04014054, 2014.

[4] N. Metni, T. Hamel, and F. Derkx, "A uav for bridge's inspection: Visual servoing control law with orientation limits," *IFAC Proceedings Volumes*, vol. 37, no. 8, pp. 454–459, 2004.

[5] I. Sa and P. Corke, "Vertical infrastructure inspection using a quadcopter and shared autonomy control," in *Field and Service Robotics*, pp. 219–232, Springer, 2014.

[6] T. Merz and F. Kendoul, "Beyond visual range obstacle avoidance and infrastructure inspection by an autonomous helicopter," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pp. 4953–4960, IEEE, 2011.

[7] C. Eschmann, C.-M. Kuo, C.-H. Kuo, and C. Boller, "High-resolution multisensor infrastructure inspection with unmanned aircraft systems," *ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, no. 2, pp. 125–129, 2013.

[8] C. Eschmann and T. Wundsam, "Web-based georeferenced 3d inspection and monitoring of bridges with unmanned aircraft systems," *Journal of Surveying Engineering*, vol. 143, no. 3, p. 04017003, 2017.

[9] S. S. Mansouri, C. Kanellakis, E. Fresk, D. Kominiak, and G. Nikolakopoulos, "Cooperative uavs as a tool for aerial inspection of the aging infrastructure," in *Field and Service Robotics*, pp. 177–189, Springer, 2018.

[10] C. Papachristos, K. Alexis, L. R. G. Carrillo, and A. Tzes, "Distributed infrastructure inspection path planning for aerial robotics subject to time constraints," in *Unmanned Aircraft Systems (ICUAS), 2016 International Conference on*, pp. 406–412, IEEE, 2016.

[11] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, "Receding horizon "next-best-view" planner for 3d exploration," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1462–1468, IEEE, 2016.

[12] C.-a. Liu, R.-f. Dong, and H. Wu, "Flying robot based viewpoint selection for the electricity transmission equipment inspection," *Mathematical Problems in Engineering*, vol. 2014, 2014.

[13] "3d robotics launches app development kit 'dronekit'." Web.

[14] "Ardupilot: Mission planner." Web.

[15] "Ardupilot: Mavproxy." Web.

[16] D. Dewey, "Splinenav - arducopter." Web, Nov 2013.

[17] O. Fernandes, R. Murphy, J. Adams, and D. Merrick, "Quantitative data analysis: Crasar small unmanned aerial systems at hurricane harvey," in *2018 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pp. 1–6, IEEE, 2018.

[18] Dronekit, "dronekit/dronekit-sitl." Web, Dec 2017.

[19] "Safe separation distance." Web.

[20] S. S. Mansouri, C. Kanellakis, D. Wuthier, E. Fresk, and G. Nikolakopoulos, "Cooperative aerial coverage path planning for visual inspection of complex infrastructures," *arXiv preprint arXiv:1611.05196*, 2016.

[21] A. Bircher, K. Alexis, M. Burri, P. Oettershagen, S. Omari, T. Mantel, and R. Siegwart, "Structural inspection path planning via iterative viewpoint resampling with application to aerial robotics," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pp. 6423–6430, IEEE, 2015.

[22] J. Heerings, N. Trimborn, and A. Den Herder, "Inspection effectiveness and its effect on the integrity of pipework," *Proceedings ECNDT*, pp. 25–29, 2006.

[23] G. Yang, R. Dong, H. Wu, and C. Liu, "Viewpoint optimization using genetic algorithm for flying robot inspection of electricity transmission tower equipment," *Chinese Journal of Electronics*, vol. 23, no. 2, pp. 426–431, 2014.

[24] D. Jones, C. Whitworth, G. Earp, and A. Duller, "A laboratory test-bed for an automated power line inspection system," *Control Engineering Practice*, vol. 13, no. 7, pp. 835–851, 2005.

[25] K. Nguyen and R. Murphy, "A survey of current practice using robotic visual assistants and automated vision planning methods,"